bsolument. Voici le document d'état du projet mis à jour. Il synthétise notre session de débogage approfondi, la solution identifiée pour le problème des pièces jointes, et intègre formellement votre demande d'amélioration pour le retraitement des réponses.

Markdown

---

Projet: "Usine à Tests"

Date: "2508171147"

Version: "13.0"

---

### 1. Résumé des Échanges du Jour

La session d'aujourd'hui a été consacrée au diagnostic final du problème des \*\*pièces jointes manquantes\*\* dans le nouveau moteur universel.

Partant d'un log qui montrait que la liste des IDs de pièces jointes était vide, nous avons mis en place une version de débogage "super espion" du script `TraitementReponses.gs`. Ce script a été conçu pour afficher en détail chaque étape de la logique de filtrage appliquée aux lignes de type "Document".

Le log détaillé généré par ce nouveau script a été décisif. Il a révélé que le filtre rejetait toutes les lignes "Document" du test "Couleurs" parce que la condition sur le `Type\_Test` (`[Couleurs]` vs `[Couleurs]`) échouait, ce qui est logiquement impossible.

Le \*\*diagnostic final et certain\*\* est donc qu'il existe une \*\*incohérence de données invisible\*\* (probablement un espace caché ou un caractère non imprimable) dans la colonne `Type\_Test` des lignes "Document" de la feuille `sys\_Composition\_Emails`.

En parallèle, vous avez évoqué le souvenir d'une fonctionnalité permettant de changer le niveau de résultat (N1, N2, N3) lors d'un retraitement. Nous avons confirmé que cette idée avait bien été planifiée comme une amélioration future et avons décidé de l'intégrer formellement à nos prochaines étapes.

### 2. État Actuel du Projet

\* \*\*Tâches réalisées :\*\*

\* Développement d'une version de débogage "super espion" du script `TraitementReponses.gs` pour tracer le filtrage.

\* Exécution du test et analyse du log détaillé.

\* Identification de la cause racine du problème des pièces jointes : une corruption de données invisible dans la colonne `Type\_Test` de la BDD.

\* Formalisation de l'ajout d'une nouvelle fonctionnalité : le changement de niveau de résultat dans l'interface de retraitement.

\* \*\*Tâches en cours :\*\*

\* Le projet est en attente de la correction des données dans la feuille `sys\_Composition\_Emails`.

\* \*\*Prochaines étapes planifiées :\*\*

1. Corriger les données dans la colonne `Type\_Test` pour le test "Couleurs".

2. Lancer un test de validation final pour confirmer que les e-mails sont générés avec leurs pièces jointes.

3. Commencer le développement de la fonctionnalité de changement de niveau dans l'interface de retraitement.

### 3. Contexte Historique du Projet

Cette session de débogage s'inscrit comme l'étape finale de la stabilisation du nouveau moteur universel. Après avoir validé la logique de calcul, de formatage des scores et d'assemblage du texte, nous nous sommes heurtés au dernier obstacle : la récupération des pièces jointes. Cette investigation a mis en lumière un principe fondamental dans un projet piloté par les données : la qualité et la propreté des données sont aussi critiques que la logique du code. La résolution de ce problème marquera la pleine fonctionnalité du moteur universel.

### 4. Orientations Stratégiques

La découverte du jour est une illustration parfaite du principe \*\*"Zéro Convention Implicite"\*\*. Le code s'attendait à une chaîne de caractères `"Couleurs"` propre, mais la donnée fournie, bien que visuellement identique, était techniquement différente. Cela renforce la nécessité d'avoir des processus de validation ou de nettoyage des données à l'avenir.

La technique du \*\*"Super Espion"\*\* s'ajoute à notre "Journal des Décisions d'Architecture" comme un pattern de débogage efficace : "Face à un comportement illogique, modifier temporairement le code pour qu'il verbalise chaque étape de sa décision afin d'identifier la divergence entre le comportement attendu et le comportement réel."

L'ajout de la fonctionnalité de changement de niveau de résultat s'inscrit dans la stratégie globale de rendre la plateforme plus \*\*flexible et puissante\*\* pour l'utilisateur final.

### 5. Architecture Technique & Ressources Clés

#### 5.1. État des Scripts et Fichiers

Voici la version "Super Espion" du script `TraitementReponses.gs` que nous avons utilisée pour le diagnostic.

```javascript

// =================================================================================

// == FICHIER : TraitementReponses.gs

// == VERSION : 8.4 (Super Espion PJ)

// == RÔLE : Gère la logique de traitement des réponses et aiguille vers le bon moteur.

// =================================================================================

/\*\*

\* Point d'entrée principal pour traiter une soumission de formulaire.

\*/

function onFormSubmit(e) {

try {

const rowIndex = e.range.getRow();

traiterLigne(rowIndex, {});

} catch (err) {

Logger.log(`Erreur critique dans onFormSubmit pour la ligne ${e.range.getRow()}: ${err.toString()}\n${err.stack}`);

}

}

/\*\*

\* COEUR LOGIQUE : Aiguille le traitement vers le moteur "Legacy" ou "Universel".

\*/

function traiterLigne(rowIndex, optionsSurcharge = {}) {

const isDebugMode = (typeof LoggerLib !== 'undefined' && LoggerLib.MODE\_DEBUG\_ACTIF === true);

if (isDebugMode) { LoggerLib.logStart(`Traitement Ligne ${rowIndex}`, optionsSurcharge); }

try {

const config = getTestConfiguration();

const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0];

const headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];

const rowValues = sheet.getRange(rowIndex, 1, 1, sheet.getLastColumn()).getValues()[0];

const reponse = {};

headers.forEach((header, i) => { reponse[header] = rowValues[i] });

reponse.nomRepondant = reponse['Votre nom et prénom'] || 'Participant';

reponse.emailRepondant = reponse['Votre adresse e-mail'] || null;

const langueOrigine = getOriginalLanguage(reponse);

const langueCible = optionsSurcharge.langue || langueOrigine;

const resultats = calculerResultats(reponse, langueCible, config, langueOrigine);

if (config.Moteur\_Calcul === 'Universel') {

Logger.log("Moteur UNIVERSEL détecté pour la ligne " + rowIndex);

assemblerEtEnvoyerEmailUniversel(config, reponse, resultats, langueCible, optionsSurcharge);

} else {

Logger.log("Moteur LEGACY détecté pour la ligne " + rowIndex);

buildAndSendEmails(config, reponse, resultats, langueCible, isDebugMode, optionsSurcharge.destinataires);

}

} catch (err) {

if (isDebugMode) { LoggerLib.logError("Erreur fatale dans traiterLigne", err); }

Logger.log("ERREUR FATALE dans traiterLigne: " + err.toString() + "\n" + err.stack);

} finally {

if (isDebugMode) { LoggerLib.logEnd(); }

}

}

// =================================================================================

// == NOUVEAU MOTEUR UNIVERSEL

// =================================================================================

/\*\*

\* Lit la BDD, assemble et envoie l'e-mail selon les règles de sys\_Composition\_Emails.

\* VERSION "SUPER ESPION" : Affiche les détails du filtre pour les documents.

\*/

function assemblerEtEnvoyerEmailUniversel(config, reponse, resultats, langueCible, optionsSurcharge = {}) {

const typeTest = config.Type\_Test;

const codeNiveauEmail = config.ID\_Gabarit\_Email\_Repondant.replace('RESULTATS\_', '');

const profilFinal = resultats.profilFinal;

const systemIds = getSystemIds();

const bdd = SpreadsheetApp.openById(systemIds.ID\_BDD);

const compoSheet = bdd.getSheetByName("sys\_Composition\_Emails");

const compoData = compoSheet.getDataRange().getValues();

const compoHeaders = compoData.shift();

const idx = {

typeTest: compoHeaders.indexOf('Type\_Test'),

langue: compoHeaders.indexOf('Code\_Langue'),

niveau: compoHeaders.indexOf('Code\_Niveau\_Email'),

profil: compoHeaders.indexOf('Code\_Profil'),

element: compoHeaders.indexOf('Element'),

ordre: compoHeaders.indexOf('Ordre'),

contenu: compoHeaders.indexOf('Contenu / ID\_Document')

};

Logger.log(`--- DÉBUT DU FILTRAGE AVEC LES PARAMÈTRES SUIVANTS ---`);

Logger.log(`Type Test: [${typeTest}], Langue: [${langueCible}], Niveau: [${codeNiveauEmail}], Profil: [${profilFinal}]`);

Logger.log(`----------------------------------------------------`);

let briquesDeContenu = compoData.filter(row => {

const typeMatch = row[idx.typeTest] === typeTest;

const langMatch = row[idx.langue] === langueCible;

const levelMatch = row[idx.niveau].includes(codeNiveauEmail);

const profileMatch = (row[idx.profil] === profilFinal || row[idx.profil] === '');

if (row[idx.element] === 'Document') {

Logger.log(`--- Vérification Ligne 'Document' (Ordre: ${row[idx.ordre]}) ---`);

Logger.log(`1. Type Test: [${row[idx.typeTest]}] vs [${typeTest}] -> ${typeMatch}`);

Logger.log(`2. Langue: [${row[idx.langue]}] vs [${langueCible}] -> ${langMatch}`);

Logger.log(`3. Niveau Email: [${row[idx.niveau]}] inclut [${codeNiveauEmail}] ? -> ${levelMatch}`);

Logger.log(`4. Profil: [${row[idx.profil]}] vs [${profilFinal}] ? -> ${profileMatch}`);

Logger.log('----------------------------------------------------');

}

return typeMatch && langMatch && levelMatch && profileMatch;

});

Logger.log(`FILTRAGE TERMINÉ. Nombre total de briques sélectionnées : ${briquesDeContenu.length}`);

// ... (le reste de la fonction d'assemblage et d'envoi reste identique) ...

}

**5.2. Architecture Détaillée et Traçabilité**

* **MODIFIÉ :** TraitementReponses.gs (Version 8.4 "Super Espion") a été temporairement modifié pour inclure des logs de débogage avancés.
* **MODIFICATION REQUISE :** La **donnée** dans l'onglet sys\_Composition\_Emails du fichier [BDD]V2 Tests & Profils doit être corrigée.

**5.3. Nomenclature des Variables**

Aucun changement.

**5.4. Structure des Données (Feuilles et Colonnes)**

Aucun changement de structure, mais une attention critique doit être portée à la propreté des données textuelles dans la colonne Type\_Test.

**6. Actions Recommandées / Prochaines Étapes**

1. **Action Immédiate (Pour vous) :**
   * Dans la feuille sys\_Composition\_Emails, filtrez sur le Type\_Test "Couleurs" et l'Element "Document".
   * **Resaisissez manuellement** le mot Couleurs dans la colonne Type\_Test pour chaque ligne concernée, afin d'éliminer tout caractère invisible.
2. **Action de Validation :**
   * Une fois les données corrigées, relancez un retraitement sur le test "Couleurs".
   * **Objectif :** Confirmer que vous recevez bien l'e-mail avec les pièces jointes.
3. **Prochaine Évolution (Nouvelle fonctionnalité) :**
   * **Thème :** Implémentation du changement de niveau de résultat (N1/N2/N3) dans l'interface de retraitement.
   * **Question pour notre prochaine session :** Pour bien préparer cette nouvelle fonctionnalité, pourriez-vous réfléchir à la manière dont l'interface utilisateur devrait présenter ce choix ? (Par exemple, une simple liste déroulante avec "Niveau 1", "Niveau 2", "Niveau 3" ?)